Theoretical Computer Science 18 (1982) 259–268
North-Holland Publishing Company

UNPROVABILITY OF THEOREMS OF COMPLEXITY THEORY IN WEAK NUMBER THEORIES

Daniel LEIVANT
Department of Computer Science, Cornell University, Ithaca, NY 14853, U.S.A.

Communicated by A. Meyer
Received November 1979
Revised September 1980

Introduction

Since 1978 there has been some interest in theorems of Computer Science that cannot be proved in certain formal theories. That such phenomena deserve study was suggested already in [6]. The recent results fall into two distinct groups. On the one hand there are properties of complex programming constructs that cannot be proved in formal theories as strong as full (impredicative) Second Order Arithmetic [15, 3, 4, 12]. The formal theories are here natural and strong, and the significance of such independence results is therefore evident. On the other hand there are theorems of complexity theory that cannot be proved in certain weak number theories [14, 1, 2, 11]. In particular, various forms of a theorem of Rabin [17] on the existence of recursive sets that are almost everywhere difficult to decide have been shown underivable in these theories [14, 11]. The significance of unprovability results of the latter kind depends on one's confidence in the intrinsic interest of the formal theories considered. The purpose of this note is to obtain the sharpest possible result on the independence of Rabin's Theorem, while at the same time to argue that the significance of the formal theories considered is limited, and that consequently the interest of such independence results lies mainly in their technical corollaries.

1. Π_2-Arithmetic

We briefly recall some notions and observations, for the most part well known.

By the *language of arithmetic* we mean the first order language with two predicate letters $=$, $<$, with 0, 1 as constants and with function symbols for all primitive recursive functions. A formula in the language of arithmetic is *true* if it is valid in the *standard model* of arithmetic whose domain is the set of natural numbers, and where each function symbol is interpreted as the primitive recursive function it

0304-3975/82/0000–0000/$02.75

denotes. Given a formula ϕ of the language of arithmetic, there are formulas ψ of the form $Q\psi_0$, where Q is a block of alternating existential and universal quantifiers, ψ_0 is quantifier free, and such that $\phi \Leftrightarrow \psi$ is provable. (By the Matijasevic–Robinson–Davis–Putnam Theorem [5] ψ_0 may be assumed to be an equation $p = 0$, where p is a polynomial.) This Normal Form Theorem gives rise to Kleene's definition of the arithmetical hierarchy, by which a formula ϕ in the language of arithmetic is Π_k or Σ_k if some ψ as above exists with k quantifiers in Q of which the first is universal or existential, respectively. A formula is *closed* if it contains no free variables. *Π_2-Arithmetic* is the first order theory in the language of arithmetic whose axioms are the true Π_2 closed formulas. Of course, this set of axioms is not recursively enumerable (r.e.).

In one sense Π_2-Arithmetic is a very strong theory: since the set of axioms is not r.e., no r.e. theory can prove all axioms of Π_2-Arithmetic. Moreover, the axioms of Π_2-Arithmetic include many major theorems and open problems of Number Theory. Fermat's 'Last Theorem', Goldbach's Conjecture and the Four Colors Theorem are actually Π_1; the existence of infinitely many twin primes and of infinitely many perfect numbers are Π_2. A number of open problems are corollaries of Riemann's Hypothesis, which itself is equivalent (modulo weak methods of proof) to a Π_2 formula [5]. Also, all theorems on consistency and relative consistency of r.e. theories may be encoded as Π_1 formulas. If $\exists x\, \phi(x)$ is a true Σ_3 closed formula, then $\phi(\bar{n})$ is a true Π_2 closed formula for some numeral $\bar{n}$; so all true Σ_3 closed formulas are theorems of Π_2-Arithmetic. The schema of induction,

$$\phi(0) \wedge \forall x\, [\phi(x) \rightarrow \phi(x+1)] \rightarrow \forall x\, \phi(x)$$

is Σ_2 when ϕ is Π_1, Σ_3 when ϕ is Σ_1; hence instances of induction for closed Π_1 and Σ_1 formulas are also theorems of Π_2-Arithmetic.

At the same time Π_2-Arithmetic is weak, since there is an instance of induction on a Σ_1 formula ϕ that cannot be proved in Π_2-Arithmetic [16, 13].

A true Π_2 closed formula is, in a sense, constructively true: if $\forall x\, \exists y\, \phi_0(x, y)$ is a true closed formula, where ϕ_0 is quantifier free, then the existence of y for each x is witnessed by a total recursive function f, i.e. $\forall x\, \phi_0(x, f(x))$ (note that this formula is not necessarily in the language of arithmetic). Conversely, every constructively true closed formula is a theorem of Π_2-Arithmetic. For if

$$\phi \equiv \forall x_1\, \exists y_1 \cdots \forall x_k\, \exists y_k\, \phi_0(x_1, \ldots, x_k, y_1, \ldots, y_k)$$

is a closed formula of the language of arithmetic with ϕ_0 quantifier free, and if there exist recursive functions that yield each y_i from $x_1, \ldots, x_i$ $(i = 1, \ldots, k)$, then there are Turing machines $M_1, \ldots, M_k$ for which the following formula ϕ^* is true:

$$\forall x_1 \ldots x_k\, \exists v_1 \ldots v_k\, \exists y_1 \ldots y_k$$
"v_1 codes a calculation of M_1 on input x_1 with y_1 as output" $\wedge \cdots \wedge$ "v_k codes a calculation of M_k on input $x_1 \ldots x_k$ with y_k as output" $\wedge \phi_0(x_1, \ldots, x_k, y_1, \ldots, y_k)$.

ϕ^*, the *constructive version of* ϕ, is a Π_2 closed formula which implies ϕ in First Order Logic. Thus, if a formula of the language of arithmetic cannot be proved in Π_2-Arithmetic, then ϕ^* is not true.

2. Non-constructiveness of a theorem of M. Rabin

2.1. Unprovability in Π_2-Arithmetic of the linear Rabin Theorem

M. Rabin [17] proved the existence, for each total recursive function h, of a recursive set R of natural numbers for which no recognizer runs infinitely often in time $O(h)$. This is easily seen to be equivalent to neither R nor $\bar{R}$ having an infinite subset I decidable in time $O(h)$, by the following argument essentially due to A.R. Meyer. For the forward implication, assume that such an I existed, say $I \subset R$; then one could run simultaneously the recognizers for I and R, accept as soon as one of the two accepts, reject when both reject; that would be a recognizer for R accepting all members of I in time $O(h)$. Conversely, given a recognizer for R that runs infinitely often in time $O(h)$, say accepting infinitely often in time $p * h(|x|)$, one could run this recognizer but reject an input x if $p * h(|x|)$ steps are made without a result; that would be a recognizer in time $O(h)$ for an infinite subset of R. We refer to the case $h(x) = |x| = \log_2(x)$ as the *linear Rabin Theorem.*

The following theorem strengthens in several respects a result of Lipton [14]. Comparison with [14] follows.

Theorem. *The linear Rabin Theorem is not a theorem of Π_2-Arithmetic.*

Proof. Suppose that the linear Rabin Theorem were provable in Π_2-Arithmetic, i.e. from a finite number of true Π_2 formulas. It is easy to see that the conjunction of a finite number of true Π_2 formulas is implied by a true Π_2 formula; thus, the linear Rabin Theorem would be provable from a certain true Π_2 formula ϕ. Without loss of generality, we assume that ϕ also implies the elementary properties of $<$.

Suppose ϕ is $\forall x\ \exists y\ \phi_0(x, y)$, where ϕ_0 is quantifier free. Let f be a total recursive function for which $\forall x\ \exists y < f(x)\ \phi_0(x, y)$. We assume without loss of generality that f is strictly increasing and that $f(x) > 2\#x$, where $\#$ denotes exponentiation. Moreover, using padding we can make $f(x)$ computable by a Turing machine M_f in time $< p * \log f(x)$ for some constant p (where $\log x$ denotes $\lceil \log_2 x \rceil$).

Let $\langle \cdots \rangle$ be a primitive recursive coding scheme for sequences of natural numbers, with a primitive recursive decoding function π satisfying $\pi(j, \langle x_0, \ldots, x_k \rangle) = x_j$ for $j = 0, \ldots, k$. (see e.g. [9]); let $\mu(k, x) \equiv \langle x, \ldots, x \rangle$ (k x's). For each integer a let I_a be the set of integers $a, f(a), f(f(a)), \ldots, f^k(a), \ldots$, where the superscript denotes the number of function compositions. Thus $x \in I_a$ is expressed by the recursive formula

$$\exists k < x\ \exists v < \mu(k, x)\ [\pi(0, v) = a \wedge \forall j < k\ \pi(j+1, v) = f(\pi(j, v)) \wedge \pi(k, v) = x]. \qquad (*)$$

We show that I_a is decidable in linear time for all a. Let M_e be a Turing machine that on input a, z simulates M_f to generate the sequence $z_0 = a$, $z_{i+1} = f(z_i)$, until some $z_j \geq \log_2 z$, or until $p * \log z$ steps are performed in the calculation of some z_i, whichever comes first. If $z_j > \log_2 z$ is reached and $z_j = z$, then M_e outputs 1, 0 otherwise. Since f is strictly increasing, $\log f(x) > x$, and M_f runs on input x in time $< p * \log f(x)$, M_e is a recognizer for $\exists i\, f^i(a) = z$, and $M_e(a, z)$ runs in time

$$\leq \sum_{i=1}^{j-1} p * \log z_i + p * \log z$$

$$\leq p * \sum_{i=1}^{\infty} \log^i z \leq m * \log z$$

where $m = 2 * p$. Also, the calculation of $M_e(a, z)$, as a list of instantaneous descriptions, can be coded by an integer v whose number of digits is $< m * \log z * ((m+1) * \log z + \log a)$, so $v < z \# (m * (m+1) * z * a)$.

For any integers a, k and Turing machine M, if M converges on $a, \ldots, f^{k-1}(a)$, then there is an integer s with a binary representation of length $k+1$ coding acceptance by M among $a, \ldots, f^{k-1}(a)$, in the sense that for $i < k$ the ith binary digit of s (starting to count from 0 with the rightmost digit) is 0 iff $M(f^i(a)) = 0$ (i.e. iff $\exists v, w\ [\pi(0, v) = a \wedge \forall j < i\ \pi(j+1, v) = f(\pi(j, v)) \wedge \{w$ codes a complete calculation of M with $\pi(i, v)$ as input and 0 as output$\}]$). We say that s is an *a,f-code* for M up to j, for each $j \leq k$. Given s as an oracle, deciding membership of $f^i(a)$ $(i < k)$ in the set R decided by M is performed in linear time by scanning s and yielding its ith digit. (We assume that the input of all Turing machines is given in binary.)

Let $\boldsymbol{L}$ denote the language of arithmetic with $+$, $*$ and $\#$ as the only function symbols, augmented with a new constant $\boldsymbol{c}$. Consider the following set Γ of formulas of $\boldsymbol{L}$, where $y = f(x)$ and similar statements are suitably interpreted by formulas of $\boldsymbol{L}$:

(1) "f is a total increasing function that majorizes a witnessing function for ϕ as well as addition, multiplication and exponentiation":

$$\forall x\, \exists y\, [y = f(x) \wedge \exists z < y\, \phi_0(x, z) \wedge f(x+1) > y \geq x \# x \geq x * x \geq x + x];$$

(2) "The function $M_e(\boldsymbol{c}, x)$ decides $I_{\boldsymbol{c}}$, and M_e runs in linear time":

$$\forall z\, \exists v < z \# (m * (m+1) * z * \boldsymbol{c})$$

"v is the code of a calculation of the Turing machine with code $\bar{e}$ on input $\boldsymbol{c}$, z,

(a) with output 1 if $z \in I_{\boldsymbol{c}}$, 0 otherwise;

(b) where the calculation's length is $< \bar{m} * \log z$.";

$(3)_k$ $\boldsymbol{c} \geq \bar{k}$;

(4) "For every Turing machine M,

$$[(\forall i < \boldsymbol{c}\ M \text{ converges on } f^i(\boldsymbol{c}))$$

$$\rightarrow$$

$$(\exists s < 2 \# (\boldsymbol{c}+2)\ s \text{ codes an } f, \boldsymbol{c}\text{-code for } M \text{ up to } \boldsymbol{c})]".$$

For each k, if $\boldsymbol{N}_k$ is the standard model with $\boldsymbol{c}$ interpreted as k, then $\boldsymbol{N}_k$ is a model of $\boldsymbol{L}$ which satisfies (1), (2), $(3)_k$ (whence also $(3)_i$ for $i<k$) and (4). The Compactness Theorem of Model Theory states that if every finite subset of a set of first order formulas has a model, then the whole set has a model. Thus there is a model $\boldsymbol{N}$ of $\boldsymbol{L}$ satisfying all formulas in Γ.

Let Z be the subset of the domain of $\boldsymbol{N}$ consisting of those elements x that are $<$ (in the sense of $\boldsymbol{N}$) than the interpretation in $\boldsymbol{N}$ of $f^i(\boldsymbol{c})$ for some (standard) integer i (i.e. for which a Σ_1 formula expressing $\exists z\ (z=f^i(\boldsymbol{c})\wedge x<z)$ is true in $\boldsymbol{N}$). Since f is strictly increasing, $x<f^i(\boldsymbol{c})$ implies $f(x)<f^{i+1}(\boldsymbol{c})$, so Z is closed under the interpretation of f in $\boldsymbol{N}$; and since (1) is true in $\boldsymbol{N}$, Z is also closed under the interpretation of $+$, $*$ and $\#$ in $\boldsymbol{N}$. The closure properties of Z imply that we may define a model $\boldsymbol{M}$ of $\boldsymbol{L}$ by restricting to Z the interpretation in $\boldsymbol{N}$ of the constant and function symbols of $\boldsymbol{L}$. Formula (1) and so also ϕ are true in $\boldsymbol{M}$ by the definition of Z. Formulas (2), $(3)_k$ and (4) can all be expressed in $\boldsymbol{L}$ as Π_1 formulas, and it is easy to see that if a Π_1 formula is true in a model, then it is true in a submodel thereof. Thus all formulas of Γ are true in $\boldsymbol{M}$.

If M is a Turing machine of $\boldsymbol{M}$ that converges on all elements of I_c, then the premise of (4) is true in $\boldsymbol{M}$. Since (4) is true in $\boldsymbol{M}$, there is in $\boldsymbol{M}$ an f, c-code s for M up to $\boldsymbol{c}$. In $\boldsymbol{M}$ if $x<f^i(c)$ and $x=f^k(\boldsymbol{c})$, then $k<i$, since f is strictly increasing in $\boldsymbol{M}$; thus, in $\boldsymbol{M}$ the set $I_{\boldsymbol{c}}$ (i.e. the extension of $(*)$) consists exactly of the elements $f^i(\boldsymbol{c})$ with i standard. Since by (3) $\boldsymbol{c}$ is non-standard, it follows that s is a code for all of $I_{\boldsymbol{c}}$, i.e. $\forall z,\ k\ [z=f^k(\boldsymbol{c})\rightarrow(M(z)=0\Leftrightarrow$ the kth digit of s is $0)]$ is true in $\boldsymbol{M}$. Hence, for each set R recursive in the sense of $\boldsymbol{M}$, membership in R among elements of $I_{\boldsymbol{c}}$ is decidable in $\boldsymbol{M}$ in linear time. In addition, the formula $\forall x\ \exists y>x\ y\in I_{\boldsymbol{c}}$, expressing that $I_{\boldsymbol{c}}$ is infinite, is true in $\boldsymbol{M}$ by the definition of Z.

Thus $\boldsymbol{M}$ is a model of the formula expressing the existence of some c for which I_c is infinite, decidable in linear time, and such that for each Turing machine recognizing a set R, membership in R among elements of I_c is decided in linear time. This formula implies the negation of the linear Rabin Theorem, since for each R at least one of $I_c\cap R$ and $I_c\cap\bar{R}$ must be infinite, and both must be decidable in linear time. The implication is proved using only the elementary properties of $<$, which we have assumed to be implied by ϕ. Thus $\boldsymbol{M}$ is a model of ϕ as well as of the negation of the linear Rabin's Theorem, and the assumption that the linear Rabin Theorem is derived from ϕ has led to a contradiction.

Corollary. *There is no recursive set R for which there exists a recursive function f such that for each Turing machine M running in linear time, either $f(M)=\langle x, y\rangle$ where x, y are accepted by M, $x\in R$ and $y\in\bar{R}$, or all numbers accepted by M are $<f(M)$.*

Proof. If a set R as above existed, then the constructive version of the linear Rabin Theorem would be true. Hence the linear Rabin Theorem would be provable in Π_2-Arithmetic, contradicting the theorem.

2.2. *Variants of Rabin's Theorem provable in Π_2-Arithmetic*

Any sublinear Rabin Theorem is provable in Π_2-Arithmetic, so the theorem above is the sharpest possible. To see this let R be the set of natural numbers whose binary representation contains an even number of 1's. Clearly, an integer must be completely scanned to decide membership in R, hence no infinite subset of R or of $\bar{R}$ can be decided in sublinear time. It is also easy to see that this argument is formalizable in Π_2-Arithmetic, and that actually the constructive version of any sublinear Rabin Theorem is a theorem of Peano's Arithmetic.

The non-constructive nature of Rabin's Theorem is dependent upon the requirement that the set R be difficult to decide almost everywhere. It *is* possible to prove in Π_2-Arithmetic a weaker variant of Rabin's Theorem to the effect that for each recursive function h there exists a recursive set R that is h-difficult to decide infinitely often (Hartmanis and Stearns [7]).

2.3. *Comparison with* [14]

In [14] Lipton considered in place of Π_2-Arithmetic the theory whose axioms are the Π_2 theorems of Peano's Arithmetic, a theory often labeled BNT, for Basic Number Theory ([8] and references there). The main result of [14] is that Rabin's Theorem for polynomial time is independent of BNT. This is weaker than our main result in that the theorem proved independent is stronger and the theory over which independence is proved is weaker.

The main theorem of [14] asserts that for each recursive R the statement that one of R, $\bar{R}$ contains an infinite subset decidable in polynomial time is unprovable in BNT. The proof of the theorem actually establishes that BNT does not prove the existence of such a set R But it should be noted that the latter independence result does not automatically follow from the statement of the theorem. To illustrate this point, consider a formula ϕ independent of BNT. Then the formula $\exists x\,[(x = 0 \wedge \phi) \vee (x = 1 \wedge \neg\phi)]$ is a theorem of BNT, although $(\bar{x} = 0 \wedge \phi) \vee (\bar{x} = 1 \wedge \neg\phi)$ is not, for each numeral $\bar{x}$.

Finally, Joseph and Young [8] have observed that the proof in [14] applies not to arbitrary recursive sets R, but only to sets that are proved in Peano's Arithmetic to be recursive.

3. The relevance of BNT and Π_2-Arithmetic to Computer Science

3.1. *The constructiveness of BNT and Π_2-Arithmetic*

We noted that if the constructive version ϕ^* of a formula ϕ is true (is provable in Peano's Arithmetic), then ϕ is a theorem of Π_2-Arithmetic (of BNT, respectively). However, the converse is generally true only for the axioms of Π_2-Arithmetic. For example, if $\psi(x)$ is a formula expressing "the xth Turing machine halts on the

empty input", then the formula

$$\forall x\ \exists y\ [(y = 0 \wedge \psi(x)) \vee (y = 1 \wedge \neg\psi(x))] \qquad (*)$$

is provable already in First Order Logic, but its constructive version expresses the recursive solvability of the halting problem, and is thus false. This nonconstructive aspect of BNT is due to the non-constructiveness of usual ('classical') First Order Logic on which BNT is based.

The existing concensus is that the elementary part of constructive reasoning is correctly captured by Heyting's (Intuitionistic) Arithmetic HA (possibly augmented by Markov's Principle; see [19]). Clearly, HA is not contained even in Π_2-Arithmetic, since all instances of induction are axioms of HA. (However, all theorems of HA without implications or negations are provable in Π_2-Arithmetic (see [19, Section 4.4.6]) though not in BNT.) Conversely, (*) is an example of a theorem of BNT which is not provable in HA. For an even simpler example consider a formula ϕ undecided in HA; then $\phi \vee \neg\phi$ is a theorem of BNT but not of HA. Thus BNT as well as Π_2-Arithmetic fail to satisfy the *disjunction property*, which is satisfied by a theory T if for each theorem of T of the form $\phi \vee \psi$ either ϕ or ψ is a theorem of T. The disjunction property is often viewed as an essential test for the constructiveness of a formal theory.

Theorems of BNT may fail to be constructive also if one understands "ϕ is constructive" to mean that ϕ is provably Π_2, i.e. $\phi \Leftrightarrow \psi$ is a theorem (of Peano's Arithmetic say) for some Π_2 formula ψ.

Proposition. *There is a theorem ϕ of Π_2-Arithmetic that is not provably equivalent to any Π_2 formula, not even in Peano's Arithmetic augmented with all true Π_1 formulas. (Moreover, ϕ is derived from the axioms of Π_2-Arithmetic using propositional inferences only.)*

Proof. Let T denote Peano's Arithmetic augmented with all true Π_1 formulas. Let Pr be a canonical Σ_2 provability predicate for T [18]. Let Con and ConCon be Π_2 formulas expressing the consistency of T and of $T + \mathrm{Con}$, respectively. Let ϕ be the formula $\mathrm{Con} \rightarrow \mathrm{ConCon}$. Write $\vdash \alpha$ for "the formula α is provable in T", and $\ulcorner\alpha\urcorner$ for the numeral for the canonical (Godel) code for α.

Arguing by contradiction, suppose that

$$\vdash \phi \Leftrightarrow \psi \qquad (1)$$

for some Π_2 formula ψ. Then

$$\vdash \neg\psi \rightarrow \mathrm{Con}. \qquad (2)$$

This implies that

$$\vdash \psi, \qquad (3)$$

by the following argument due to Kreisel [10].

We use the derivability conditions (relativized to T) for Pr (see [18, p. 827]). Since $\neg\psi$ is Σ_2,

$$\vdash \neg\psi \to \mathrm{Pr}(\ulcorner\neg\psi\urcorner), \tag{4}$$

by the second derivability condition. By the first and third conditions, (2) implies

$$\vdash \mathrm{Pr}(\ulcorner\neg\psi\urcorner) \to \mathrm{Pr}(\ulcorner\mathrm{Con}\urcorner). \tag{5}$$

Finally, by Godel's Second Incompleteness Theorem for T,

$$\vdash \mathrm{Pr}(\ulcorner\mathrm{Con}\urcorner) \to \neg\mathrm{Con}. \tag{6}$$

Combining (4), (5) and (6),

$$\vdash \neg\psi \to \neg\mathrm{Con},$$

which with (2) implies (3).

But (1) and (3) imply that $\mathrm{Con} \to \mathrm{ConCon}$ is a theorem of T, i.e. that the theory $T + \mathrm{Con}$ proves its own consistency, contradicting Godel's Second Incompleteness Theorem.

3.2. On the thesis that BNT is adequate as a canonical formal system for Computer Science

The kinship of true Π_2 and constructively true formulas has led Lipton [14] to put forward the thesis that BNT is an adequate canonical formal theory for Computer Science. We propose that the reader assess the merits of this suggestion with in mind the observations of Section 3.1 about non-constructive aspects of BNT, and the following remarks concerning the strength of BNT and the epistemological consistency of Lipton's Thesis.

We mentioned that there are Σ_1 formulas for which induction is not provable even in Π_2-Arithmetic. This is a weakness that we suspect few computer scientists would accept of their canonical formal system. The absence in Π_2-Arithmetic of weak instances of induction is also reflected by severe anomalies of models of Π_2-Arithmetic. As an example, consider the following application of the Completeness Theorem, sometimes dubbed 'the overspill principle' and due in this context to Joseph and Young [8], to show that such a model may contain a set S that is (in the sense of the model) finite, r.e., but not recursive.

Let T be a formal theory and $\phi(x, y)$ a formula of the language of arithmetic. Assume that T proves the elementary properties of addition. By the Completeness Theorem for First Order Logic T proves induction on ϕ with respect to x,

$$\phi(0, y) \wedge \forall x\,[\phi(x, y) \to \phi(x+1, y)] \to \forall x\,\phi(x, y),$$

iff for each model $\boldsymbol{M}$ of T and each element y of $\boldsymbol{M}$, the set $D_{\phi,y}$ of elements x of $\boldsymbol{M}$ for which $\phi(x, y)$ is true in $\boldsymbol{M}$ either is unbounded in $\boldsymbol{M}$ or contains a last element. Let $\phi(x, y)$ be a Σ_1 formula for which induction with respect to x is not

provable in Π_2-Arithmetic [16, 13]. Thus, there is a model $\boldsymbol{M}$ of Π_2-Arithmetic and an element y of $\boldsymbol{M}$ such that the set $S = D_{\phi,y}$ is bounded but has no last element. Since induction over Π_1 formulas *is* provable in Π_2-Arithmetic, S cannot be equal to $D_{\psi,z}$ for some z and recursive relation $\psi(x, z)$, i.e. S is not recursive in the sense of $\boldsymbol{M}$. Joseph and Young [8] showed that models of Π_2-Arithmetic may be plagued by even more exotic maladies.

Furthermore, the suggestion that BNT be adopted as canonical system seems to us epistemologically inconsistent. BNT is proposed as an alternative to Peano's Arithmetic because of the alleged non-constructive nature of those proofs in Peano's Arithmetic that use complex formulas. At the same time, many formulas are justified as axioms of BNT precisely by these rejected proofs. To illustrate this point, let n be some large integer, and consider the fragment PA_n of Peano's Arithmetic in which induction is allowed only on Π_n formulas. Then the formula Con_n expressing the consistency of PA_n is Π_1; also, Con_n can be proved in Peano's Arithmetic, but only by using induction on a Π_{n+1} formula. Thus, Con_n is an axiom of BNT, whose simplest proof in Peano's Arithmetic is highly 'non-constructive'. Should Lipton's Thesis be accepted, a typical formal proof in Computer Science would consist of two parts, of which the first would provide derivations in Peano's Arithmetic for axioms of BNT, using arbitrarily complex formulas, and the second would derive some theorems from these axioms.

3.3. The significance of independence results over BNT and Π_2-Arithmetic

It has been suggested [2, 8] that independence results over BNT and Π_2-Arithmetic are valuable in calibrating the proof theoretic complexity of theorems of Computer Science. The most useful classification of proofs of Peano's Arithmetic is by the complexity of instances of induction they use (also useful are classifications by length and other structural properties, which are irrelevant here). However, if ϕ is a formula independent of BNT, then all we know is that ϕ cannot have a proof in Peano's Arithmetic of the particular form mentioned at the end of Section 3.2 above; among these excluded proofs are arbitrarily complex ones, but not all proofs that use induction only on Σ_1 formulas. We therefore doubt that independence over BNT is a promising classification principle. Unprovability of ϕ in BNT does exclude a proof of ϕ by induction on closed Π_1 and Σ_1 formulas only, but this is already excluded by ϕ being underivable in the much weaker theory PA_1.

It seems that independence results over BNT or Π_2-Arithmetic teach us something about the non-constructiveness of the independent formula rather than of its potential proofs. If ϕ is found unprovable in Π_2-Arithmetic, then ϕ is not constructively true (as illustrated by the admittedly undramatic corollary above). Similarly, if ϕ is independent of BNT, then no recursive Skolem functions witnessing its prenex form (as in Section 1 above) can be proved in Peano's Arithmetic to witness ϕ (though this does not even guarantee that the witnessing functions are not provably recursive in Peano's Arithmetic).

Acknowledgment

We gratefully acknowledge constructive criticism by Albert R. Meyer and the referee of an earlier version of this note.

References

[1] R. DeMillo and R. Lipton, Some connections between computational complexity theory and mathematical logic, *Proc. 11th Annual ACM Symposium on Theory of Computing* (1979) 153–159.
[2] R. DeMillo and R. Lipton, The consistency of "$N = NP$" and related problems with fragments of number theory, *Proc. 12th Annual ACM Symposium on Theory of Computing* (1980) 45–57.
[3] S. Fortune, *Topics in Complexity Theory*, PhD Dissertation, Cornell University (1979).
[4] S Fortune, D. Leivant and M. O'Donnell, The expressiveness of simple and second order type structures, *J. ACM*, to appear.
[5] M. Davis, Y. Matijasevic and J. Robinson, Hilbert's tenth problem, *Proc. Symposia Pure Math.* **28** (1976) 328–378.
[6] J. Hartmanis and J. Hopcroft, Independence results in Computer Science, *SIGACT News* **8** (4) (1976) 13–23.
[7] J. Hartmanis and R.E. Stearns, On the computational complexity of algorithms, *Trans. Amer. Math. Soc.* **117** (1965) 285–306.
[8] D. Joseph and P. Young, Independence results in computer science?, *Proc. 12th Annual ACM Symposium on Theory of Computing* (1980) 58–69.
[9] S.C. Kleene, *Introduction to Metamathematics* (Wolters-Noordhoff, Groningen, 1952).
[10] G. Kreisel, Review of: Szabo, Ed., *The Collected Papers of Gerhard Gentzen*, *J. Philosophy* (1971) 238–265.
[11] D. Leivant, On easily infinite sets and on a statement of R. Lipton, Report TR 79-390, Department of Computer Science, Cornell University (1979).
[12] D. Leivant, The complexity of argument passing in polymorphic procedures, *Proc. 13th Annual Symposium on Theory of Computing* (1981) 38–45.
[13] D. Leivant, The optimality of induction as an axiomatization of arithmetic, *J. Symbolic Logic*, to appear.
[14] R.J. Lipton, Model theoretic aspects of complexity theory, *Proc. 19th Symposium on the Foundations of Computer Science* (1978) 176–188.
[15] M. O'Donnell, A programming language theorem which is independent of Peano Arithmetic, *Proc. 11th ACM Symposium on Thoery of Computing* (1979) 176–188.
[16] C. Parsons, On a number theoretic choice schema and its relation to induction, in: J. Myhill, A. Kino and R.E. Vesley, Eds., *Intuitionism and Proof Theory* (North-Holland, Amsterdam, 1970) 459–474.
[17] M.O. Rabin, Degree of difficulty of computing a function and a partial ordering of recursive sets, Technical Report 2, The Hebrew University, Jerusalem (1960).
[18] C. Smoryński, The incompleteness theorems, in: J. Barwise, Ed., *Handbook of Mathematical Logic* (North-Holland, Amsterdam, 1977) 821–866.
[19] A.S. Troelstra, *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis* (Springer, Berlin, 1974).